Building the knowledge base of the question-answer system based on the syntagmatic analysis of the text

A Zarubin[1] and A Koval[1]

[1] The Bonch-Bruevich Saint - Petersburg State University of Telecommunication, Saint - Petersburg, Russia

Abstract. Question-answer (QA) systems are systems that can take questions and respond to them in a natural language. In most cases, the principles of building question-answer systems are used in the development of decision support systems. The mechanism of syntagmatic patterns is used when processing open-ended questions and when extracting answers to it from semi-structured resources. This article describes the application of the mechanisms of syntagmatic patterns in the construction of various types of QA-systems and expert systems.

1. Introduction

There is currently a problem of rapid access to information. This problem occurs, for example, when:

- interaction with customers;
- making managerial decisions;
- technical support services, etc

Question-answer systems can be used to solve the problem of operational access to information [1, 2, 3, 4, 5, 6]. Question-answer systems generate a response based on an analysis of the user question. The UIMA architecture is currently used to build question-answer systems [4, 5, 6, 7]. Modern question-answer systems show significant results [4, 5, 6], but they require the presence of highly skilled specialists for tuning and training [7, 8, 9].

This article describes an attempt to develop a question-answer system using syntagmatic patterns [10]. Syntagmatic pattern is a template for detecting certain syntagmatic units in the text. Syntagmatic unit is a collection of several words united on the principle of semantic-grammatical-phonetic compatibility. Relations between syntagmatic units are taken into account when using syntagmatic patterns. For example, the syntagmatic pattern "building * the knowledge base" will allow you to find sentences containing syntagmatic units:

- building a knowledge base;
- building a corporate knowledge base;
- building a fuzzy knowledge base, etc.

Our approach is based on the following ideas:

1. The knowledge base of the question-answer system can be generated automatically based on the analysis of unstructured text resources.
2. The use of syntagmatic patterns to organize the structure of the knowledge base of the question-answer system allows one to effectively search for answers to questions.

2. A structure of the KB of the QA system

The knowledge base (KB) of our question-answer system has a tree-like structure. Semantic networks are currently actively used in the construction of a KB [8]. Formally, the structure of the KB looks like this:

$$KB = \langle SP, TD, R \rangle,$$

where $SP = \{SP_1, SP_2, \text{K}, SP_n\}$ is a set of syntagmatic patterns;

$TD = \{TD_1, TD_2, \text{K}, TD_n\}$ – is a set of text data (KB content);

$R = \{R^{SP}, R^{TD}\}$ – is a set of relations of KB:

$R^{SP} = \{R_1^{SP}, R_2^{SP}, \text{K}, R_n^{SP}\}$ – is a set of relations between the internal nodes of the KB tree;

$R^{TD} = \{R_1^{TD}, R_2^{TD}, \text{K}, R_n^{TD}\}$ – is a set of relations between the internal and terminal nodes of the KB tree.

The internal nodes of the KB tree contain a syntagmatic pattern as a label. Terminal nodes contain text information. The answer to the question will be extracted from this textual information. An example of the KB structure of our question-answer system is shown in Figure 1.

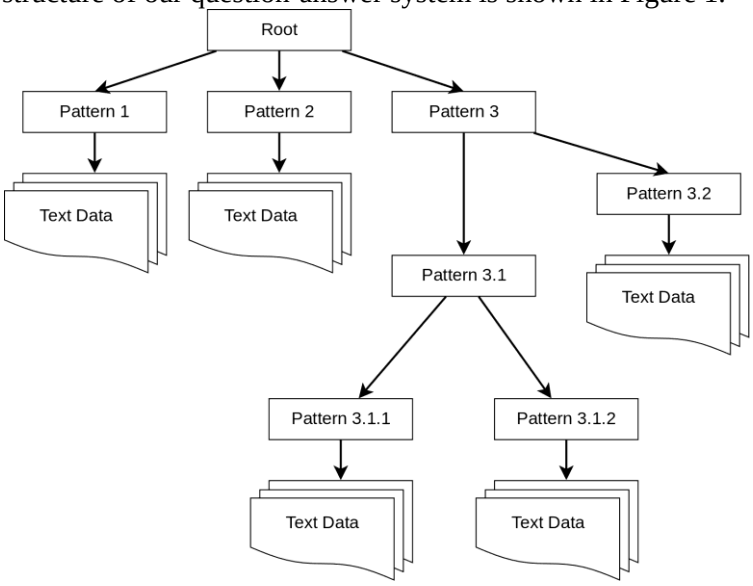

Figure 1. The structure of the knowledge base of our question-answer system.

More general syntagmatic patterns are located closer to the root element of the tree. More precise syntagmatic patterns are located closer to the terminal nodes of the tree. Thus, this knowledge base structure of our question-answer system allows us to find the necessary terminal nodes on the user's question (if the answer to this question is in the knowledge base).

3. Learning of the QA system KB

The modified fuzzy C-Means (FCM) fuzzy clustering algorithm is used for learning (building a tree) the KB of our QA system. It is necessary to present each document as an index for the FCM algorithm. Indexing documents consists of the following steps:

1. Download the document.
2. Removing stop words (words that do not have semantic value: prepositions, particles, etc.).
3. Stemming using the Porter algorithm (highlighting the basis of the word).
4. Calculation the frequency of occurrence of words in the document.
5. The index of the document can be represented as an expression:

$$I_d = \left\{\left(w_1^d, \mathrm{f_i^d}\right), \left(w_2^d, \mathrm{f_2^d}\right), \mathrm{K}, \left(w_n^d, \mathrm{f_n^d}\right)\right\},$$

where w_i^d – i-th word of the document d;

f_i^d – is the frequency of occurrence of i-th word in the document d;

n – is the number of words in the document d.

The modified FCM clustering algorithm is based on minimizing the function:

$$F^{FCM} = \sum_{i=1}^{D}\sum_{j=1}^{C} u_{ij}^m \left\| I_i - I_j^c \right\|^2, 1 \le m \le \infty,$$

where D – is the number of document indexes for clustering;

C – is a number of clusters;

m – is any real number greater than 1;

u_{ij} – is the degree to which the document index belongs I_i to the cluster j;

I_i – i-th document index;

I_j^c – is a center of j-th cluster;

$\left\| I_i - I_j^c \right\|$ – the normalized distance between the index of the document and the center of the cluster.

The FCM algorithm consists of the following steps:

1. Initialization of the matrix of indexes belonging to documents to clusters:
$$U = \lfloor u_{ij} \rfloor.$$
2. Calculation of cluster centers:
$$I_j^c = \frac{\sum_{i=1}^{D} u_{ij}^m \cdot I_i}{\sum_{i=1}^{D} u_{ij}^m}.$$
3. Formation of a new membership matrix:
$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\left\| I_i - I_j^c \right\|}{\left\| I_i - I_k^c \right\|} \right)^{\frac{2}{m-1}}}.$$
4. The value of the objective function is calculated. The obtained value is compared with the value at the previous iteration. Clustering is complete if the difference does not exceed the threshold value. Otherwise, go to the second step of the algorithm.

The knowledge base is learned in the process of hierarchical clustering. First, the entire set of document indexes I^0 is clustered. Clusters are formed after the algorithm is executed. Each cluster obtained contains a subset of the documents of the original set: $I^1 \cup I^2 \cup \mathrm{K} \cup I^n = I^0$. A new partition is performed for each cluster received. The split continues as long as the value $\sqrt{D/2}$ is greater than or equal to 2.

Thus, a tree is constructed whose internal nodes contain indexes of cluster centers, and terminal nodes contain text data. It is necessary to form internal node labels in the form of syntagmatic patterns based on the contents of internal nodes.

4. An algorithm for constructing syntagmatic patterns

There are currently many approaches to the analysis of texts in natural language [1, 11, 12, 13, 14, 15]. Statistical and/or linguistic methods for the analysis of texts in natural language underlie such approaches. Methods for the analysis of texts on natural language are also used in the development of QA systems [1, 2, 3, 5, 6, 7].

The analysis of texts on natural language consists of the following steps:

1. Grafematic analysis is the selection of structural elements of the text (sentences, names, dates, etc.).
2. Morphological analysis is the definition of the morphological features of the words of the sentence (part of speech, gender, etc.).
3. Parsing is the selection of the syntactic units of the sentence (subject, predicate, etc.).
4. Semantic analysis is the definition of the meaning of the sentence.

It is sufficient to use the first two steps of the process of text analysis in natural language to form syntagmatic patterns: graphematic and morphological analysis.

The algorithm for syntagmatic patterns can be written as follows in pseudocode:

For each terminal node of the knowledge base.
For each document from the current terminal node.
Split the text of the document into sentences.
For each proposal of the document.
Split the sentence into words.
For each word of the sentence.
Get morphological signs.
For each untreated internal node, except for the root node.
Until the root node is reached or the parent of the current node is not the root node.
For each word of the index of the parent ejected node.
Find all the documents in which the word appears.
For each document.
Select syntagmatic units.
Delete the same syntagmatic units.
For each selected syntagmatic unit.
Find similar syntagmatic units.
Form candidates in syntagmatic patterns.
Solve the problem of optimal coverage.
Write the result in the node label.
Mark the node as processed.
Go to the parent node.

Thus, internal nodes of the knowledge base tree are marked with syntagmatic patterns as a result of this algorithm.

5. The search for the answer to the question in the KB

The learned KB of our QA system allows us to find answers to the user's requests. First you need to find the required terminal node of the knowledge base. The internal node labels are used to find the most relevant terminal node. Each internal node of the knowledge base is marked with a syntagmatic pattern.

In pseudocode, the search algorithm for the most relevant terminal node of the knowledge base tree can be written as follows:

Syntagmatic units are allocated from the question that has entered the system entrance.
Set the position on the root node of the knowledge base tree.
For each child node of the knowledge base.
Find the number of syntagmatic patterns of the current child node
corresponding to the syntagmatic units of the question.
Set the position to the child node with the highest match.
Continue if the selected child node is not terminal, otherwise, return the terminal node.

It is necessary to find in the text documents the most relevant sentence after finding the terminal node. The answer to the question is the most relevant sentence.

In pseudocode, the search algorithm for the most relevant sentence from the text documents of the terminal node found can be written as follows:

For each terminal node document.
For each sentence of the current document.

Find the number of syntagmatic patterns of the current sentence corresponding to the syntagmatic units of the question.
Choose the best match.

Select the document with the highest match.

Thus the two algorithms presented above make it possible to organize the search for the most relevant answer to an incoming question.

6. Experiments

The materials of the Sberbank Data Science Contest [16] were used as data for experiments. These materials contain 50,365 entries of the form "paragraph, question, answer." The answer to the question is always the exact text substring of the paragraph, with precision to punctuation and the text register. Each paragraph contains several sentences.

Two question-answer systems were used to conduct the experiment. Their knowledge base was learned using a modified FCM algorithm. The first knowledge base contained many pairs of "term-frequency" as labels of internal nodes. The second knowledge base contained syntagmatic patterns as labels of internal nodes.

A proximity measure was used to find the most relevant document and / or sentence in the first knowledge base. The proximity measure is obtained using the square of the Euclidean distance:

$$Dist(I_i^c, I^q) = \sum_{w=1}^{W} \left(f_w^c - f_w^q \right)^2,$$

where I_i^c – is the index of the i-th terminal node document c;

I^q – is the index of the received question;

W – is the number of words in the index I_i^c;

f_w^c, f_w^q – is the frequency of occurrence of the word w in the indexes I_i^c and I^q.

The most relevant document and / or sentence is a document and / or sentence with a minimum proximity measure.

During the experiment 50,365 questions were submitted to both question-answer systems. The sentence from the paragraph was given as an answer to the question. The result was considered successful if the reference answer was a substring of the found sentence. The results of the experiments are presented in Table 1.

Table 1. The comparison of statistical and syntagmatic approaches to the implementation of the question-answer system.

Type of KB	Number of errors	Percent of errors
Vector	23178	46,0
Syntagmatic	9723	19,3

As can be seen from the results of the experiments, the syntagmatic approach to the implementation of the question-answer system made it possible to reduce the number of errors from 46% to 19.3%.

7. Conclusion

Thus, the developed syntagmatic approach to the development of question-answer systems is effective. This approach can be used to develop the following types of software systems:

- the system for automating the process of interaction with customers based on the analysis of the knowledge base and corporate correspondence;
- decision support system based on the analysis of the knowledge base and use cases;
- the system of verification of information flows of the enterprise to ensure information security;

- the system for automating the work of the technical support service based on the analysis of the knowledge base and use cases.

In the future, we plan to modify the developed approach by finding answers to questions in an implicit form.

8. References

[1] Jurafsky, D., Martin J.H., Speech and Language Processing. Available at: https://web.stanford.edu/~jurafsky/slp3/28.pdf (accessed: 03.05.2018)

[2] Berant, J. Semantic parsing on freebase from question-answer pairs / Berant, J., Chou, A., Frostig, R., Liang, P. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2013. pp 1533–1544.

[3] Bordes, A., Chopra, S., Weston, J. Question answering with subgraph embeddings Available at: https://arxiv.org/pdf/1406.3676.pdf (accessed: 03.05.2018)

[4] Epstein, E.A., Schor, M.I., Iyer, B., Lally, A., Brown, E.W., Cwiklik, J. Making watson fast / IBM Journal of Research and Development. 2012. Vol. 56(3.4). pp 15–19.

[5] Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A.A., Lally, A., Murdock, J.W., Nyberg, E., Prager, J., Et Al. Building watson: An overview of the deepqa project / AI magazine. 2010. Vol. 31(3). pp 59–79.

[6] Gallagher, S., Zadrozhny, W., Shalaby, W., Avadhani, A. Watsonsim: Overview of a question answering engine. Available at: https://arxiv.org/pdf/1412.0879.pdf (29.04.2018)

[7] Ferrucci, D., Lally, A. UIMA: An architectural approach to unstructured information processing in the corporate research environment / Nat. Lang. Eng. 2004. Vol. 10(3-4). pp. 327–348.

[8] Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J. Freebase: a collaboratively created graph database for structuring human knowledge / Proceedings of the 2008 ACM SIGMOD international conference on Management of data. 2008. pp 1247–1250.

[9] Chen, D., Manning, C. D. fast and accurate dependency parser using neural networks / Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014. pp 740–750.

[10] Zarubin, A., Koval, A., Filippov, A., Moshkin, V. Application of syntagmatic patterns to evaluate answers to open-ended questions / Proceedings of the 2017 Communications in Computer and Information Science (CITDS). 2017. pp 150–162.

[11] Zarubin A.A., Koval A.R., Moshkin V.S., Filippov A.A. Construction of the problem area ontology based on the syntagmatic analysis of external wiki-resources. Available at: http://ceur-ws.org/Vol-1903/paper26.pdf (accessed: 04.05.2018)

[12] Boyarskiy K.K., Kanevskiy Ye.A. Semantic and syntactic parser SemSin / Nauchno-tekhnicheskiy vestnik informatsionnykh tekhnologiy, mekhanikiioptiki. – 2015. Vol. 5. pp. 869-876.

[13] Artemov M.A., Vladimirov A.N., Seleznev K. E. Review of Russian NLP systems. Available at: http://www.vestnik.vsu.ru/pdf/analiz/2013/02/2013-02-31.pdf (03.11.2017)

[14] Automatic text processing. Available at: http://aot.ru (accessed: 29.04.2018)

[15] Lally, A., Prager, J.M., McCord, M.C., Boguraev, B., Patwardhan, S., Fan, J., Fodor, P., Chu-Carroll, J. Question analysis: How watson reads a clue / IBM Journal of Research and Development. 2012. Vol. 56(3.4) (2012). pp 2–14.

[16] Sberbank Data Science Contest. Available at: https://contest.sdsj.ru (accessed: 04.05.2018)

Acknowledgments

This paper has been approved within the framework of the federal target project “R&D for Priority Areas of the Russian Science-and-Technology Complex Development for 2014-2020”, government contract No 14.607.21.0164 on the subject “The development of architecture, methods and models to build software and hardware complex semantic analysis of semi-structured information resources on the Russian element base” (Application Code « 2016-14-579-0009-0687 »).